CSE- 4029 LAB Assignment - 3

Academic year: 2021-2022 **Semester:** WIN

Faculty Name: Prof. BKSP Kumar raju Alluri sir **Date:** 1/4/2022

Student name: M.Taran **Reg. no.:** 19BCE7346

Logistic Regression

Step-1 : Took Dataset and imported libraries

importing essential libraries

library(knitr)

library(tidyverse)

library(ggplot2)

library(DataExplorer)

link for dataset :

https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

Importing Datasets

Clients_data <- read.csv("D:/users/lenovo/Downloads/Client_data.csv", header=TRUE)

View(Clients_data)

data

Filter

	ID	BALANCE	GENDER	STUDY	SPLICED	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2
1	1	20000	2	2	1	24	2	2	-1	-1	-2	-2	3913	3
2	2	120000	2	2	2	26	-1	2	0	0	0	2	2682	17
3	3	90000	2	2	2	34	0	0	0	0	0	0	29239	140
4	4	50000	2	2	1	37	0	0	0	0	0	0	46990	482
5	5	50000	1	2	1	57	-1	0	-1	0	0	0	8617	56
6	6	50000	1	1	2	37	0	0	0	0	0	0	64400	570
7	7	500000	1	1	2	29	0	0	0	0	0	0	367965	4120
8	8	100000	2	2	2	23	0	-1	-1	0	0	-1	11876	3
9	9	140000	2	3	1	28	0	0	2	0	0	0	11285	140

Step-2 : Exploratory Data Analysis(EDA)

dim(clients_Data)

```
> dim(clients_Data)
[1] 30000    25
>
```

head(clients_Data)

```
> head(clients_Data)
  ID BALANCE GENDER STUDY SPLICED AGE PAY_0 PAY_2
1  1   20000      2     2       1  24     2     2
2  2  120000      2     2       2  26    -1     2
3  3   90000      2     2       2  34     0     0
4  4   50000      2     2       1  37     0     0
5  5   50000      1     2       1  57    -1     0
6  6   50000      1     1       2  37     0     0

  PAY_3 PAY_4 PAY_5 PAY_6 BILL_AMT1 BILL_AMT2
1    -1    -1    -2    -2      3913      3102
2     0     0     0     2      2682      1725
3     0     0     0     0     29239     14027
4     0     0     0     0     46990     48233
5    -1     0     0     0      8617      5670
6     0     0     0     0     64400     57069
```

str(clients_Data)

```
> str(clients_Data)
'data.frame':   30000 obs. of  25 variables:
 $ ID                           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ BALANCE                      : num  20000 120000 90000 50000 50000 50000
0000 100000 140000 20000 ...
 $ GENDER                       : int  2 2 2 2 1 1 1 2 2 1 ...
 $ STUDY                        : int  2 2 2 2 2 1 1 2 3 3 ...
 $ SPLICED                      : int  1 2 2 1 1 2 2 2 1 2 ...
 $ AGE                          : int  24 26 34 37 57 37 29 23 28 35 ...
 $ PAY_0                        : int  2 -1 0 0 -1 0 0 0 0 -2 ...
 $ PAY_2                        : int  2 2 0 0 0 0 0 -1 0 -2 ...
 $ PAY_3                        : int  -1 0 0 0 -1 0 0 -1 2 -2 ...
 $ PAY_4                        : int  -1 0 0 0 0 0 0 0 0 -2 ...
 $ PAY_5                        : int  -2 0 0 0 0 0 0 0 0 -1 ...
 $ PAY_6                        : int  -2 2 0 0 0 0 0 -1 0 -1 ...
```

clients_Data[, 1:25] <- sapply(clients_Data[, 1:25], as.character)

clients_Data[, 1:25] <- sapply(clients_Data[, 1:25], as.numeric)

str(clients_Data)

```
> str(clients_Data)
'data.frame':   30000 obs. of  25 variables:
 $ ID                           : num  1 2 3 4 5 6 7 8 9 10 ...
 $ BALANCE                      : num  20000 120000 90000 50000 50000 50000
0000 100000 140000 20000 ...
 $ GENDER                       : num  2 2 2 2 1 1 1 2 2 1 ...
 $ STUDY                        : num  2 2 2 2 2 1 1 2 3 3 ...
 $ SPLICED                      : num  1 2 2 1 1 2 2 2 1 2 ...
 $ AGE                          : num  24 26 34 37 57 37 29 23 28 35 ...
 $ PAY_0                        : num  2 -1 0 0 -1 0 0 0 0 -2 ...
 $ PAY_2                        : num  2 2 0 0 0 0 0 -1 0 -2 ...
 $ PAY_3                        : num  -1 0 0 0 -1 0 0 -1 2 -2 ...
 $ PAY_4                        : num  -1 0 0 0 0 0 0 0 0 -2 ...
 $ PAY_5                        : num  -2 0 0 0 0 0 0 0 0 -1 ...
 $ PAY_6                        : num  -2 2 0 0 0 0 0 -1 0 -1 ...
```

summary(clients_Data)

```
> summary(clients_Data)
       ID           BALANCE             GENDER          STUDY      
 Min.   :    1   Min.   :  10000   Min.   :1.000   Min.   :0.000  
 1st Qu.: 7501   1st Qu.:  50000   1st Qu.:1.000   1st Qu.:1.000  
 Median :15000   Median : 140000   Median :2.000   Median :2.000  
 Mean   :15000   Mean   : 167484   Mean   :1.604   Mean   :1.853  
 3rd Qu.:22500   3rd Qu.: 240000   3rd Qu.:2.000   3rd Qu.:2.000  
 Max.   :30000   Max.   :1000000   Max.   :2.000   Max.   :6.000  
```

introduce(clients_Data)

```
> introduce(clients_Data)
   rows columns discrete_columns continuous_columns all_missing_columns
1 30000      25                0                 25                   0
  total_missing_values complete_rows total_observations memory_usage
1                    0         30000             750000      6005808
>
```

count(clients_Data, vars = STUDY)

```
> count(clients_Data, vars = STUDY)
  vars     n
1    0    14
2    1 10585
3    2 14030
4    3  4917
5    4   123
6    5   280
7    6    51
>
```

count(clients_Data, vars = SPLICED)

```
> count(clients_Data, vars = SPLICED)
  vars     n
1    0    54
2    1 13659
3    2 15964
4    3   323
>
```

#replace 0's with NAN, replace others too

clients_Data$STUDY[clients_Data$STUDY == 0] <- 4

clients_Data$STUDY[clients_Data$STUDY == 5] <- 4

clients_Data$STUDY[clients_Data$STUDY == 6] <- 4

clients_Data$SPLICED[clients_Data$SPLICED == 0] <- 3

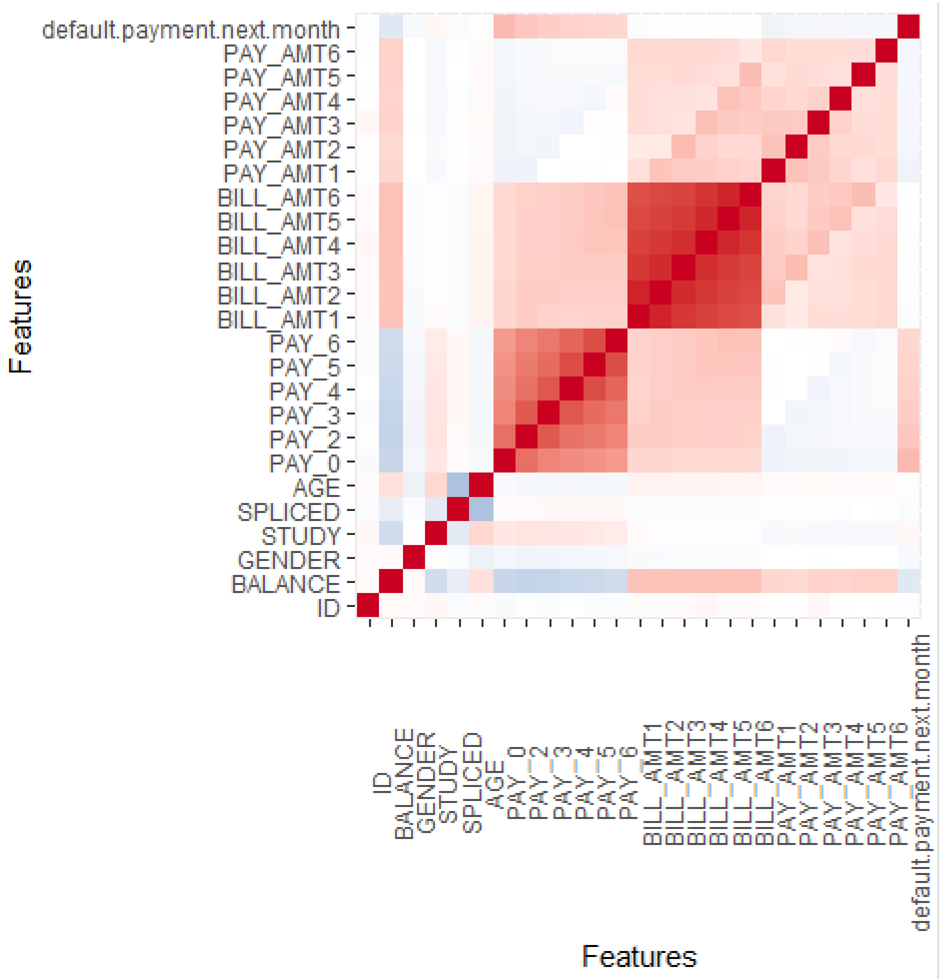
count(clients_Data, vars = SPLICED)

```
> count(clients_Data, vars = SPLICED)
  vars     n
1    1 13659
2    2 15964
3    3   377
>
```

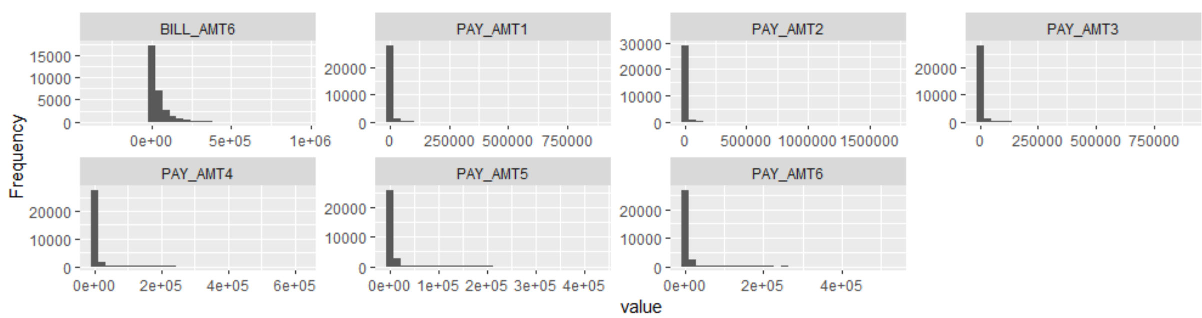
count(clients_Data, vars = STUDY)

```
> count(clients_Data, vars = STUDY)
  vars     n
1    1 10585
2    2 14030
3    3  4917
4    4   468
```

plot_correlation(na.omit(clients_Data), maxcat = 5L)



plot_histogram(clients_Data)



Step-3: Feature Engineering

#deleting columns

clients_Data_new <- select(clients_Data, -one_of('ID','AGE', 'BILL_AMT2', 'BILL_AMT3','BILL_AMT4','BILL_AMT5','BILL_AMT6'))

head(clients_Data_new)

```
> head(clients_Data_new)
  BALANCE GENDER STUDY SPLICED PAY_0 PAY_2
1   20000      2     2       1     2     2
2  120000      2     2       2    -1     2
3   90000      2     2       2     0     0
4   50000      2     2       1     0     0
5   50000      1     2       1    -1     0
6   50000      1     1       2     0     0

  PAY_3 PAY_4 PAY_5 PAY_6 BILL_AMT1 PAY_AMT1
1    -1    -1    -2    -2      3913        0
2     0     0     0     2      2682        0
3     0     0     0     0     29239     1518
4     0     0     0     0     46990     2000
5    -1     0     0     0      8617     2000
6     0     0     0     0     64400     2500
```

Step-4: Pre-processing

clients_Data_new[, 1:17] <- scale(clients_Data_new[, 1:17])

head(clients_Data_new)

```
> head(clients_Data_new)
     BALANCE     GENDER      STUDY    SPLICED
1 -1.1367012  0.8101472  0.2118664 -1.0687794
2 -0.3659744  0.8101472  0.2118664  0.8491164
3 -0.5971924  0.8101472  0.2118664  0.8491164
4 -0.9054832  0.8101472  0.2118664 -1.0687794
5 -0.9054832 -1.2343024  0.2118664 -1.0687794
6 -0.9054832 -1.2343024 -1.1313270  0.8491164

        PAY_0     PAY_2      PAY_3      PAY_4
1  1.79453395 1.7823185 -0.6966518 -0.6665876
2 -0.87497656 1.7823185  0.1388625  0.1887429
3  0.01486028 0.1117342  0.1388625  0.1887429
4  0.01486028 0.1117342  0.1388625  0.1887429
5 -0.87497656 0.1117342 -0.6966518  0.1887429
6  0.01486028 0.1117342  0.1388625  0.1887429
```

splitting the clients_Data for training and testing

#create a list of random number ranging from 1 to number of rows from actual clients_Data

#and 70% of the clients_Data into training clients_Data clients_Data2 = sort(sample(nrow(clients_Data_new), nrow(clients_Data_new)*.7))

```
clients_Data2    int [1:21000] 1 2 3 4 5 7 9 10 13 14 ...
```

#creating training clients_Data set by selecting the output row values

train <- clients_Data_new[clients_Data2,]

```
train           21000 obs. of 18 variables
```

#creating test clients_Data set by not selecting the output row values

test <- clients_Data_new[-clients_Data2,]

```
test            9000 obs. of 18 variables
```

dim(train)

dim(test)

```
> dim(train)
[1] 21000    18
> dim(test)
[1] 9000   18
>
```

Step-5: Model Building

fit a logistic regression model with the training dataset

log.model <- glm(default_payment ~., clients_Data = train, family = binomial(link = "logit"))

summary(log.model)

```
> log.model <- glm(default.payment.next.month ~., data = train, family = binomial(link = "logit"))
> summary(log.model)

Call:
glm(formula = default.payment.next.month ~ ., family = binomial(link = "logit"),
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1160  -0.7030  -0.5504  -0.2913   3.4350
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept) -1.45218    0.01966 -73.848  < 2e-16 ***
BALANCE     -0.09691    0.02410  -4.021 5.81e-05 ***
GENDER      -0.04987    0.01772  -2.815 0.004883 ** 
STUDY       -0.06039    0.01920  -3.146 0.001655 ** 
SPLICED     -0.13129    0.01829  -7.179 7.01e-13 ***
PAY_0        0.64081    0.02370  27.039  < 2e-16 ***
PAY_2        0.10939    0.02867   3.815 0.000136 ***
PAY_3        0.09010    0.03209   2.808 0.004987 ** 
PAY_4        0.03857    0.03504   1.101 0.270962    
PAY_5        0.02262    0.03623   0.624 0.532497    
PAY_6        0.02059    0.02973   0.693 0.488463    
BILL_AMT1   -0.13135    0.02366  -5.550 2.85e-08 ***
PAY_AMT1    -0.12381    0.03552  -3.486 0.000491 ***
PAY_AMT2    -0.23107    0.05465  -4.228 2.35e-05 ***
PAY_AMT3    -0.05341    0.03191  -1.674 0.094189 .  
PAY_AMT4    -0.05668    0.02885  -1.965 0.049448 *  
PAY_AMT5    -0.04032    0.02703  -1.492 0.135721    
PAY_AMT6    -0.04207    0.02579  -1.631 0.102807    
```

#Step-6: Prediction

test[1:10,]

```
> test[1:10,]
      BALANCE     GENDER      STUDY    SPLICED       PAY_0      PAY_2      PAY_3      PAY_4
6  -0.9054832 -1.2343024 -1.1313270  0.8491164  0.01486028  0.1117342  0.1388625  0.1887429
8  -0.5201198  0.8101472  0.2118664  0.8491164  0.01486028 -0.7235579 -0.6966518  0.1887429
11  0.2506070  0.8101472  1.5550597  0.8491164  0.01486028  0.1117342  1.8098911  0.1887429
12  0.7130431  0.8101472 -1.1313270  0.8491164 -0.87497656 -0.7235579 -0.6966518 -0.6665876
21 -0.2889017  0.8101472  1.5550597  0.8491164  0.01486028  0.1117342  0.1388625  0.1887429
24  2.1774240  0.8101472 -1.1313270 -1.0687794 -1.76481340 -1.5588500 -1.5321662 -1.5219182
25 -0.5971924 -1.2343024 -1.1313270  0.8491164  0.01486028  0.1117342  0.1388625 -0.6665876
30 -0.9054832 -1.2343024 -1.1313270  0.8491164  0.01486028  0.1117342  0.1388625  0.1887429
33 -0.5201198 -1.2343024 -1.1313270  0.8491164  0.01486028  0.1117342  0.1388625  0.1887429
34  2.5627874  0.8101472  0.2118664 -1.0687794 -1.76481340 -1.5588500 -1.5321662 -1.5219182

        PAY_5      PAY_6  BILL_AMT1     PAY_AMT1   PAY_AMT2     PAY_AMT3   PAY_AMT4    PAY_AMT5
6   0.2349126  0.2531332  0.1789436 -0.190999635 -0.1782122 -0.259481541 -0.2442256 -0.24867860
8   0.2349126 -0.6164414 -0.5343501 -0.318993604 -0.2309012 -0.296796327 -0.2709711 -0.20371288
11  0.2349126 -0.6164414 -0.5452551 -0.202712291 -0.2564644 -0.293956542 -0.2889079 -0.06947024
12 -0.6475540  1.9922823 -0.5291217  0.975315225  0.1755505  0.190681311  1.1154567 -0.31413088
21  0.2349126 -0.6164414 -0.1747156 -0.160812378 -0.1902777 -0.240000610 -0.1803937 -0.25326026
24 -1.5300205 -1.4860160 -0.6207754  0.831020136 -0.1930554 -0.264990726 -0.3080574 -0.31413088
25  0.2349126  0.2531332 -0.6312051  0.005640157 -0.2569852  0.009786953 -0.2314592 -0.18028096
30  0.2349126  0.2531332 -0.4874572 -0.251374149 -0.1918835 -0.240000610 -0.2442256 -0.20940723
33  0.2349126  0.2531332  0.5678302 -0.159423764 -0.1046038 -0.109256870 -0.1035402 -0.10468357
34 -1.5300205 -1.4860160 -0.5472107 -0.091260938  0.7337325  0.130364260  4.2520263 -0.24992219
```

to predict using logistic regression model, probabilities obtained

log.predictions <- predict(log.model, test, type="response")

```
log.predictions    Large numeric (9000 elements,  648.2 kB)
```

Look at probability output

head(log.predictions, 10)

```
> head(log.predictions, 10)
         6          8         11         12         21         24         25 
0.23068128 0.18425025 0.20841400 0.07902886 0.17032774 0.04828542 0.23365811 
        30         33         34 
0.24842430 0.20756812 0.02631928 
>
```

log.prediction.rd <- ifelse(log.predictions > 0.5, 1, 0)

head(log.prediction.rd, 10)

```
> head(log.prediction.rd, 10)
 6  8 11 12 21 24 25 30 33 34 
 0  0  0  0  0  0  0  0  0  0 
```

#Step-7: Model Evaluation

table(log.prediction.rd, test[,18])

```
> table(log.prediction.rd, test[,18])

log.prediction.rd    0    1
                0 6858 1493
                1  184  465
```

accuracy <- table(log.prediction.rd, test[,18])

sum(diag(accuracy))/sum(accuracy)

```
> accuracy <- table(log.prediction.rd, test[,18])
> sum(diag(accuracy))/sum(accuracy)
[1] 0.8136667
>
```